**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**

**УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ**

**ГОМЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ИМЕНИ П. О. СУХОГО**

Факультет автоматизированных и информационных систем

Кафедра «Информатика»

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 2

по дисциплине «Разработка приложений для Интернет»

на тему: «Работа с файловой системой и потоками в Node.js»

Выполнил: студент гр. ИП-31

Коваленко А.И.

Принял: преподаватель

Самовендюк Н. В.

Гомель 2023

**Цель:** изучить основные возможности по работе с файловой системой и потоками в Node.js

**Задание 1.**

Написать приложение, которое логирует ошибки (вводятся пользователем) в текстовом файле. В файл добавляется дата ошибки и ее описание.

**Задание 2.**

Создать массив объектов на основе класса в соответствии со своим вариантом. Записать этот массив в формате JSON в файл.

|  |  |  |
| --- | --- | --- |
| 15 | Фитнес-браслет | Объем оперативной памяти  Поддерживаемые платформы  Датчики |

**Задание 3.**

Произвести считывание данных из созданного файла в новый массив. Вывести массив в консоль.

**Задание 4.**

Создать в папке проекта две папки src и dist. В папку src скопировать исходные файлы приложений (\*.js). В папку dist скопировать созданные файлы.

**Листинг программы:**

**App1.js**

const { newFile, writeErrors } = require('./Library');

const readline = require('readline');

const rl = readline.createInterface({

    input: process.stdin,

    output: process.stdout

})

const fileName = "FirstTask.txt";

rl.question(`Please, enter your error: `, answ => {

    newFile(fileName);

    writeErrors(fileName, answ);

    rl.close();

})

**App2.js**

const { CreateFitnessTracker, writeToJSON} = require('./Library');

const fileName = 'SecondTask.json';

let Trackers = CreateFitnessTracker();

writeToJSON(fileName, JSON.stringify(Trackers, null, '\t'));

**App3.js**

const fs = require('fs');

fs.readFile("SecondTask.json", function(err, data) {

if (err) throw err;

const box = JSON.parse(data);

console.log(box);

});

**App4.js**

const fs = require('fs');

const path = require('path');

files = fs.readdirSync('./');

console.log("\Filenames with the .js extension:");

function MakeDir()

{

fs.mkdir('dist', (err) => {

if (err)

console.log(err);

});

fs.mkdir('src', (err) => {

if (err)

console.log(err);

});

}

function CopyJsFile()

{

fs.copyFile(file, `.\\src\\${file}`, err => {

if (err) throw err;

console.log('Файл успешно скопирован');

});

}

function CopyNotJsFile()

{

fs.copyFile(file, `.\\dist\\${file}`, err => {

if (err) throw err;

console.log('Файл успешно скопирован');

});

}

function CopyFile() {

fs.mkdir('dist', (err) => {

if (err)

console.log(err);

});

fs.mkdir('src', (err) => {

if (err)

console.log(err);

});

files.forEach(file => {

if (path.extname(file) == ".js") {

fs.copyFile(file, `.\\src\\${file}`, err => {

if (err) throw err;

console.log('Файл успешно скопирован');

});

}

else {

fs.copyFile(file, `.\\dist\\${file}`, err => {

if (err) throw err;

console.log('Файл успешно скопирован');

});

}

})

}

CopyFile()

**Library.js**

const fs = require("fs"), path = require('path');

const newFile = (fileName) => {

fs.access(fileName, function(error){

if (error) {

fs.writeFile(fileName, '', (err) => {

if (err)

console.log (err);

else

console.log('file was created succesfully');

});

} else {

console.log("file was found");

}

});}

const writeErrors = (fileName, error) => {

fs.appendFile(fileName, `\n${error} ${new Date()}`, (err)=>{

if (err)

console.log (err);

else

console.log ('file was changed');

});}

const writeToJSON = (fileName, strings) => {

fs.writeFile(fileName, strings, (err) => {

if (err)

console.log (err);

else

console.log('file was created succesfully');

});

}

class FitnessTracker{

constructor(ram, support, sensor){

this.ram = ram;

this.support = support;

this.sensor = sensor;

}

get RAM(){

return this.ram;

}

set RAM(rams){

this.ram = rams;

}

get Support(){

return this.support;

}

set Support(supports){

this.support = supports;

}

get Sensors(){

return this.sensor;

}

set RAM(sensors){

this.sensor = sensors;

}

}

function CreateFitnessTracker(){

let arrayTracker = [];

arrayTracker[0] = new FitnessTracker(

"1024",

"ios",

"Accelerometer"

);

arrayTracker[1] = new FitnessTracker(

"256",

"android",

"Gyroscope"

);

arrayTracker[2] = new FitnessTracker(

"512",

"mi",

"Magnetometer"

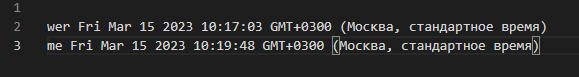
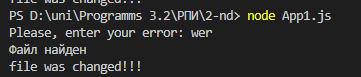
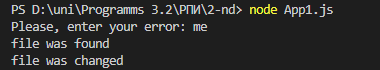
);

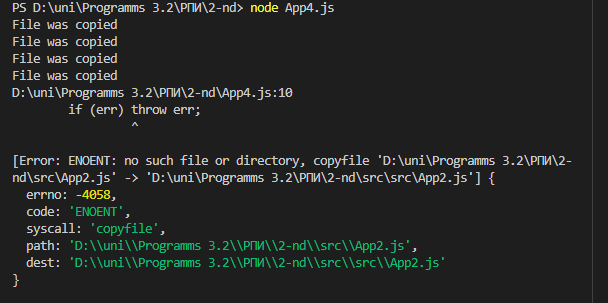
return arrayTracker;

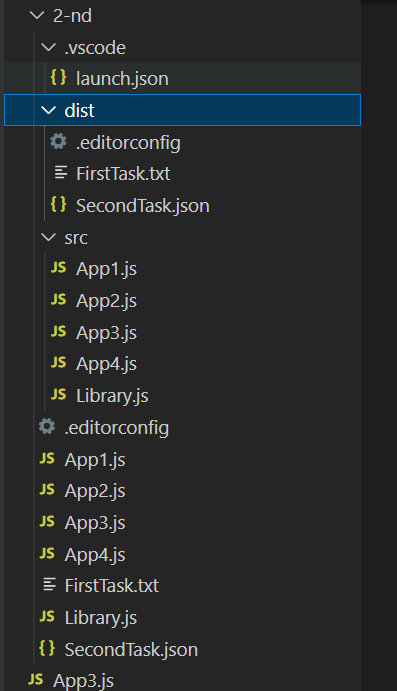
}

module.exports = { fs, newFile, writeErrors, FitnessTracker, CreateFitnessTracker, writeToJSON }

**Результаты выполнения программы:**

****

****



**Вывод:** в ходе выполнения лабораторной работы были изучены основные возможности по работе с файловой системой и потоками в Node.js.